```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;
contract BankAccount {
    address public owner;
    uint256 private balance;
    // Events for logging transactions
    event Deposit(uint256 amount, uint256 newBalance);
    event Withdrawal(uint256 amount, uint256 newBalance);
    // Set the owner of the contract to the account deploying it
    constructor() {
        owner = msg.sender;
        balance = 0;
    }
    // Modifier to restrict access to the owner only
    modifier onlyOwner() {
        require(msg.sender == owner, "Only the account owner can perform this action");
        _;
    }
    // Function to deposit money into the account
    function deposit() public payable onlyOwner {
        require(msg.value > 0, "Deposit amount must be greater than zero");
        balance += msg.value;
        emit Deposit(msg.value, balance);
    }
    // Function to withdraw money from the account
    function withdraw(uint256 amount) public onlyOwner {
        require(amount <= balance, "Insufficient balance");
        balance -= amount;
        payable(msg.sender).transfer(amount);
        emit Withdrawal(amount, balance);
    }
    // Function to show the current balance of the account
    function showBalance() public view returns (uint256) {
        return balance;
    }
}
```